

AMENDMENTS TO THE CLAIMS:

Please amend the claims as follows. The claims are in the format as required by 35 C.F.R. § 1.121.

1. (Currently amended) A system for interfacing applications comprising a computer executable software ~~architecture~~ instructions stored on a computer readable memory, said computer executable software ~~architecture~~ further comprising:
instructions operable to implement:

a public application program interface ("API") (public API) for a set of heterogeneous workflow engines, wherein ~~the said public API~~ comprises a set of generic objects; and

a plurality of adapters, each adapter configured to interface with a workflow engine application program interface (workflow engine API), wherein each workflow engine API is associated with an underlying workflow engine ~~from the of said set of~~ heterogeneous workflow engines;

wherein each adapter is operable to map said set of generic objects to a set of native objects for a corresponding underlying workflow engine.

2. (Original) The system of Claim 1, wherein said set of generic objects comprises a generic process definition object and wherein each of said plurality of adapters is operable to translate said generic process definition object into a native process definition object for said corresponding underlying workflow engine.

3. (Original) The system of Claim 1, wherein said set of generic objects further comprises a payload object.

4. (Original) The system of Claim 3, wherein said payload object associates a set of content items with a process instance.

5. (Currently amended) The system of Claim 1, wherein in said set of generic objects is maintained based upon an industry standard for workflow management.
6. (Currently amended) The system of Claim 5, wherein said ~~set of generic objects is based upon a~~ industry standard is a reference model promulgated by the Workflow Management Coalition.
7. (Currently amended) The system of Claim 6, wherein ~~said a generic object from said set of generic software objects is one of~~ comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and ~~or~~ a WfRole object.
8. (Original) The system of Claim 7, wherein said set of generic objects further comprises a WfPayload object.

9. (Currently amended) A system for interfacing workflow applications comprising a computer executable software ~~architecture-instructions~~ stored on a computer readable ~~memory-medium~~, said computer executable software architecture further comprising: instructions operable to implement:

a first workflow engine;

a first workflow engine application program interface (workflow engine API) associated with said first workflow engine, wherein said first workflow engine API comprises a first set of native objects;

a second workflow engine;

a second workflow engine API associated with said second workflow engine, wherein said second workflow engine API comprises a second set of native objects, wherein ~~the-said~~ first workflow engine and said second workflow engine are heterogeneous workflow engines;

a public application program interface (public API) comprising a set of generic objects for the heterogeneous workflow engines;

a first adapter configured to map said set of generic objects to said first set of native objects; and

a second adapter configured to map said set of generic ~~object-objects~~ objects to said second set of native objects.

10. (Original) The system of Claim 9, wherein said set of generic objects comprises a generic process definition object and wherein said first adapter is operable to translate said generic process definition object into to a first native process definition object and wherein said second adapter is operable to translate said generic process definition object into a second native process definition object.

11. (Previously Presented) The system of Claim 9, wherein said set of generic objects further comprises a payload object.

12. (Previously Presented) The system of Claim 11, wherein said payload object associates a set of content items with a process instance.

13. (Currently amended) The system of Claim 9, wherein ~~in~~ said set of generic objects is maintained based upon an industry standard for workflow management.

14. (Currently amended) The system of Claim 13, wherein said ~~set of generic objects is based upon a~~ industry standard is a reference model promulgated by the Workflow Management Coalition.

15. (Currently amended) The system of Claim 14, wherein ~~said a generic object from said set of generic software objects is one of~~ comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and ~~or~~ a WfRole object.

16. (Previously Presented) The system of Claim 15, wherein said set of generic objects further comprises a WfPayload object.

17. (Currently amended) A computer-implemented method for integrating workflow engines comprising:

creating a public application program interface (public API) for at least two heterogeneous underlying workflow engines, wherein the public API comprises a set of generic objects,

wherein said set of generic objects represent functional characteristics common to said at least two heterogeneous underlying workflow engines,

wherein each of the at least two heterogeneous underlying workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition, and

wherein each of said at least two heterogeneous underlying workflow engines has an associated application program interface (workflow engine API) and a set of native objects;

interfacing said public API with ~~the~~ said at least two heterogeneous underlying workflow engines through ~~an~~ said associated workflow engine API for each of the said at least two heterogeneous underlying workflow engines; and

mapping said set of generic objects to ~~a set of~~ native objects for ~~of~~ each of said at least two heterogeneous underlying workflow engines.

18. (Currently amended) The method of Claim 17, further comprising:

persistently maintaining a generic process definition object; and

delegating at least a portion of ~~the~~ said set of generic objects to a set of corresponding native objects at one or more of said underlying workflow engines.

19. (Previously Presented) The method of Claim 18, further comprising translating said generic process definition object into a native process definition object and persistently maintaining said native process definition object.

20. (Currently amended) The method of Claim 17, further comprising:
receiving a call from an application understandable by a generic object from said set of generic objects;
mapping said ~~to-call~~ from said application to a native call understandable by a native object from said set of native objects;
~~prosecuting-executing~~ said native call to generate a native result; and
mapping said native result to a generic result usable by a generic object from said set of generic objects.
21. (Previously Presented) The method of Claim 20, wherein said set of generic objects further comprises a payload object.
22. (Previously Presented) The method of Claim 21, wherein said payload object associates a set of content items with a process instance.
23. (Currently amended) The method of Claim 20, wherein ~~in~~ said set of generic objects is maintained based upon an industry standard for workflow management.
24. (Currently amended) The method of Claim 23, wherein said ~~set of generic objects is based upon a~~ industry standard is a reference model promulgated by the Workflow Management Coalition.
25. (Currently amended) The method of Claim 24, wherein ~~said a generic object from said set of generic software objects is one of~~ comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and ~~or~~ a WfRole object.
26. (Previously Presented) The method of Claim 25, wherein said set of generic objects further comprises a WfPayload object.

27. (New) A computer-implemented method for providing a standardized application program interface between a plurality of software applications and a plurality of workflow engines, said method comprising:

- creating and maintaining a public application program interface (public API) comprising a set of generic software objects;

- wherein said set of generic software objects represent functional characteristics common to at least two heterogeneous workflow engines;

- wherein each of said at least two heterogeneous workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition;

- wherein each of said at least two heterogeneous workflow engines has an application program interface (workflow engine API) associated therewith;

- wherein each workflow engine API comprises a set of native software objects; and

- wherein workflow engine APIs of said at least two heterogeneous workflow engines are incompatible; and

- translating and mapping said set of generic software objects of said public API to and from said set of native software objects of said each workflow engine API, facilitating interoperability of said plurality of software applications.